

What is claimed is:

1. In a computer system, a method for managing services associated with a plurality of plug-in modules, the method comprising the steps of:

5           obtaining identities of a plurality of plug-in modules;  
          retrieving a dependency list indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of a plurality of plug-in modules;  
          calculating a plug-in initiation order based upon the dependency list indicating  
10          respective plug-in services provided by, and required by, each plug-in module; and  
          initiating service operation of plug-in modules according to the plug-in initiation  
          order, such that if a first plug-in module provides a service required by a second plug-in  
          module, the first plug-in module is initiated such that the service provided by the first  
          plug-in module is available to the second plug-in module when required by the second  
15          plug-in module.

2. The method of claim 1 wherein the step of obtaining identities of a plurality of plug-in modules includes the steps of:

20           receiving a list of services to be started within the computerized device;  
          determining, for each service in the list of services, a respective plug-in module  
          definition that can provide that service; and  
          placing the identity of each plug-in module definition determined in the step of  
          determining into the identities of the plurality of plug-in modules.

25          3. The method of claim 1 wherein the step of retrieving a dependency list indicating respective plug-in services provided by, and required by, each plug-in module comprises the steps of:

          for each plug-in module identified in the identities of a plurality of plug-in modules, performing the steps of:

instantiating the plug-in module based upon a plug-in module definition associated with the identity of the plug-in module;

receiving a dependency response from the plug-in module, the dependency response indicating respective plug-in services provided by, and required by, the plug-in module; and

5 storing identities of the respective plug-in services provided by, and required by, the plug-in module as identified in the dependency response in the dependency list.

10 4. The method of claim 3 wherein the step of instantiating the plug-in module comprises the steps of:

obtaining plug-in initiation information corresponding to the plug-in module definition associated with the identity of the plug-in module;

instantiating the plug-in module based upon a plug-in module definition

15 associated with the identity of the plug-in module; and

passing the plug-in initiation information to the plug-in module for use by the plug-in module.

20 5. The method of claim 3 wherein the step of instantiating the plug-in module comprises the step of:

querying a dependency interface associated with the plug-in module with a dependency query to obtain the dependency response from the plug-in module.

25 6. The method of claim 1 wherein the step of calculating a plug-in initiation order based upon the dependency list comprises the step of:

arranging a placement of each plug-in module identified in the dependency list within the plug-in initiation order such that plug-in modules not requiring services provided by other plug-in modules are placed earlier in the initiation order and such that plug-in modules requiring services provided by other plug-in modules are placed later in 30 the initiation order.

7. The method of claim 6 wherein the step of arranging placement of each plug-in module identified in the dependency list within the plug-in initiation order comprises the steps of:

5       analyzing the dependency list indicating respective plug-in services provided by, and required by, each plug-in module to determine which plug-ins provide services relied upon by other plug-in modules; and

10       creating, as the plug-in initiation order, at least one plug-in module dependency tree based on the step of analyzing, the at least one plug-in module dependency tree containing a hierarchical arrangement of nodes associated with respective plug-in modules, the hierarchical arrangement indicating the plug-in initiation order of the plug-in modules respectively associated with the nodes in the dependency tree.

15       8. The method of claim 7 wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order comprises the steps of:

20       traversing the at least one plug-in module dependency tree according to the hierarchical arrangement of nodes and for each node encountered during the step of traversing, initiating service operation of the respective plug-in module associated with that node.

25       9. The method of claim 8 wherein the step of initiating service operation of the respective plug-in module associated with that node comprises the steps of:

      forwarding, via a dependency available interface associated with the respective plug-in module, a list of initiated plug-in services of other plug-in modules that are currently available for use by the respective plug-in module.

10. The method of claim 1 wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order comprises performing, for each respective plug-in module in the plug-in initiation order, the steps of:

determining, from a published list of services available by initiated plug-in modules, an identity of each initiated plug-in service required by the respective plug-in module;

5 forwarding to the respective plug-in module, via a dependency available interface associated with the respective plug-in module, the identity of each initiated plug-in service required by the respective plug-in module;

receiving a list of services initiated by the respective plug-in module; and

10 adding the list of services provided by the respective plug-in module to the published list of services.

11. The method of claim 1, wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order operates such that if the second plug-in module requires a service provided by the first plug-in module, the second plug-in module is initiated such that the service provided by the first plug-in module is available 15 to the second plug-in module when required.

12. The method of claim 1 wherein the first plug-in module is initiated via the step of initiating operation of plug-in modules prior to initiation of the second plug-in module.

20 13. The method of claim 1 wherein the first plug-in module is initiated via the step of initiating operation of plug-in modules after initiation of the second plug-in module, and wherein the second plug-in module includes a wait-state operation causing the second plug-in module to wait to provide the service offered by the second plug-in module until initiation of the first plug-in module.

25 14. The method of claim 1 wherein the steps of obtaining, retrieving, calculating and initiating are performed by a multi-threaded plug-in manager and wherein the step of calculating a plug-in initiation order is performed by collectively operating a respective thread for each plug-in, each thread performing the step of initiating service operation of

at least one plug-in module when all services required by that plug-in module are available.

15. A computer system comprising:

- 5        a memory;
- a processor; and
- an interconnection mechanism coupling the memory and the processor;
- wherein the memory is encoded with a plug-in manager application that, when performed on the processor, produces a plug-in manager process that manages services
- 10      associated with a plurality of plug-in modules encoded within the memory by performing the operation steps of:
  - obtaining identities of a plurality of plug-in modules in the memory;
  - retrieving, into the memory, a dependency list indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of
  - 15      a plurality of plug-in modules;
  - calculating, in the memory, a plug-in initiation order based upon the dependency list indicating respective plug-in services provided by, and required by, each plug-in module; and
  - initiating service operation of plug-in modules on the processor according to the
  - 20      plug-in initiation order, such that if a first plug-in module provides a service required by a second plug-in module, the first plug-in module is initiated such that the service provided by the first plug-in module is available to the second plug-in module when required by the second plug-in module.
  - 25      16. The computer system of claim 15 wherein when the plug-in manager process performs the step of obtaining identities of a plurality of plug-in modules, the plug-in manager process performs the steps of:
    - receiving a list of services to be started within the computerized device;
    - determining, for each service in the list of services, a respective plug-in module
    - 30      definition that can provide that service; and

placing the identity of each plug-in module definition determined in the step of determining into the identities of the plurality of plug-in modules.

17. The computer system of claim 15 wherein when the plug-in manager process  
5 performs the step of retrieving a dependency list indicating respective plug-in services provided by, and required by, each plug-in module, the plug-in manager process performs the steps of:

for each plug-in module identified in the identities of a plurality of plug-in modules, performing the steps of:

10 instantiating the plug-in module in the memory based upon a plug-in module definition associated with the identity of the plug-in module;

receiving a dependency response from the plug-in module, the dependency response indicating respective plug-in services provided by, and required by, the plug-in module; and

15 storing, in the memory, identities of the respective plug-in services provided by, and required by, the plug-in module as identified in the dependency response in the dependency list.

18. The computer system of claim 17 wherein when the plug-in manager process  
20 performs the step of instantiating the plug-in module, the plug-in manager process performs the steps of:

obtaining, in the memory, plug-in initiation information corresponding to the plug-in module definition associated with the identity of the plug-in module;

25 instantiating the plug-in module in the memory based upon a plug-in module definition associated with the identity of the plug-in module; and

passing the plug-in initiation information to the plug-in module in the memory for use by the plug-in module.

19. The computer system of claim 17 wherein when the plug-in manager process performs the step of instantiating the plug-in module, the plug-in manager process performs the step of:

querying a dependency interface associated with the plug-in module with a

5 dependency query to obtain the dependency response from the plug-in module.

20. The computer system of claim 15 wherein when the plug-in manager process performs the step of calculating a plug-in initiation order based upon the dependency list, the plug-in manager process performs the step of:

10 arranging a placement of each plug-in module identified in the dependency list within the plug-in initiation order such that plug-in modules not requiring services provided by other plug-in modules are placed earlier in the initiation order and such that plug-in modules requiring services provided by other plug-in modules are placed later in the initiation order.

15

21. The computer system of claim 20 wherein when the plug-in manager process performs the step of arranging placement of each plug-in module identified in the dependency list within the plug-in initiation order, the plug-in manager process performs the steps of:

20 analyzing the dependency list indicating respective plug-in services provided by, and required by, each plug-in module to determine which plug-ins provide services relied upon by other plug-in modules; and

25 creating in the memory, as the plug-in initiation order, at least one plug-in module dependency tree based on the step of analyzing, the at least one plug-in module dependency tree containing a hierarchical arrangement of nodes associated with respective plug-in modules, the hierarchical arrangement indicating the plug-in initiation order of the plug-in modules respectively associated with the nodes in the dependency tree.

22. The computer system of claim 21 wherein when the plug-in manager process performs the step of initiating service operation of plug-in modules according to the plug-in initiation order, the plug-in manager process performs the step of:

traversing the at least one plug-in module dependency tree in the memory  
5 according to the hierarchical arrangement of nodes and for each node encountered during the step of traversing, initiating service operation of the respective plug-in module associated with that node.

23. The computer system of claim 22 wherein when the plug-in manager process  
10 performs the step of initiating service operation of the respective plug-in module associated with that node, the plug-in manager process performs the step of:

forwarding, via a dependency available interface associated with the respective plug-in module, a list of initiated plug-in services of other plug-in modules that are currently available for use by the respective plug-in module.

15  
24. The computer system of claim 15 wherein when the plug-in manager process performs the step of initiating service operation of plug-in modules according to the plug-in initiation order the plug-in manager process performs, for each respective plug-in module in the plug-in initiation order, the steps of:

20 determining, from a published list of services available by initiated plug-in modules, an identity of each initiated plug-in service required by the respective plug-in module;

forwarding to the respective plug-in module, via a dependency available interface associated with the respective plug-in module, the identity of each initiated plug-in  
25 service required by the respective plug-in module;

receiving a list of services initiated by the respective plug-in module; and

adding the list services provided by the respective plug-in module to the published list of services.

25. The computer system of claim 15, wherein the step of initiating service operation of plug-in modules according to the plug-in initiation order operates in the plug-in manager process such that if the second plug-in module requires a service provided by the first plug-in module, the second plug-in module is initiated such that the service provided by  
5 the first plug-in module is available to the second plug-in module when required.

26. The computer system of claim 15 wherein the plug-in manager initiates the first plug-in module via the step of initiating operation of plug-in modules prior to initiation of the second plug-in module.

10

27. The computer system of claim 15 wherein the plug-in manager process initiates the first plug-in module via the step of initiating operation of plug-in modules after initiation of the second plug-in module, and wherein the second plug-in module includes a wait-state operation causing the second plug-in module to wait to provide the service offered  
15 by the second plug-in module until initiation of the first plug-in module.

28. The computer system of claim 15 wherein the steps of obtaining, retrieving, calculating and initiating are performed by a multi-threaded plug-in manager and wherein the step of calculating a plug-in initiation order is performed by collectively operating a  
20 respective thread for each plug-in, each thread performing the step of initiating service operation of at least one plug-in module when all services required by that plug-in module are available.

29. A computer program product having a computer-readable medium including  
25 computer program logic encoded thereon, that when executed on a computer system having a coupling of a memory and a processor, provides a plug-in manager process for managing plug-in services by causing the processor to perform the operations of:  
obtaining identities of a plurality of plug-in modules in the memory;

retrieving, into the memory, a dependency list indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of a plurality of plug-in modules;

5 calculating, in the memory, a plug-in initiation order based upon the dependency list indicating respective plug-in services provided by, and required by, each plug-in module; and

10 initiating service operation of plug-in modules on the processor according to the plug-in initiation order, such that if a first plug-in module provides a service required by a second plug-in module, the first plug-in module is initiated such that the service provided by the first plug-in module is available to the second plug-in module when required by the second plug-in module.

30. A computer system comprising:

15 a memory;

a processor; and

an interconnection mechanism coupling the memory and the processor;

20 wherein the memory is encoded with a plug-in manager application that, when performed on the processor, produces a plug-in manager process that manages services associated with a plurality of plug-in modules encoded within the memory by operating on the computer system and causing the computer system to provide:

means for obtaining identities of a plurality of plug-in modules in the memory;

means for retrieving, into the memory, a dependency list indicating respective plug-in services provided by, and required by, each plug-in module identified in the identities of a plurality of plug-in modules;

25 means for calculating, in the memory, a plug-in initiation order based upon the dependency list indicating respective plug-in services provided by, and required by, each plug-in module; and

30 means for initiating service operation of plug-in modules on the processor according to the plug-in initiation order, such that if a first plug-in module provides a service required by a second plug-in module, the first plug-in module is initiated such

that the service provided by the first plug-in module is available to the second plug-in module when required by the second plug-in module.